



PRACTICAL WORK N°2

Basic use of the Command Line under GNU/Linux

➔ Rappel !



☒ Kind of use OS :

WSL (Windows Sub-system for Linux) ; **Virtual Machine** ; **Dual-boot** setup ; **Live DVD** ; **clef USB live**

☒ Terminology :

GUI (Graphical-User Interface) ; **CLI** (Command-Line Interface)

Shell ; **CLI** (Command-Line Interpreter) ; **Terminal** ; **Command Prompt** ; **Windows PowerShell**

home / homedirectory ; root ; command completion ; wildcard (like * or ?)

☒ Documentation :

It is not possible to memorize all the parameters and arguments of a command.

This is why the **Linux** system offers important documentation on the various commands in the form of a **manual**, accessible from the **shell** via the **man** or **xman** command.

1. Run the command : **man man** and read the document in the **manual**.

2. **Using the manual pages** : All the manual pages are organized in the same way.

- NAME : Command Name ;
- SYNOPSIS : Short summary of the command syntax ;
- DESCRIPTION: long or short description of the command;
- OPTIONS (optional): description of the supported options;
- You can of course add as many sub-sections as you like;
- AUTHOR: the people who developed the command;
- SEE ALSO: Cross-references.

☒ Most famous and popular shells :

- **Bourne Shell** and its derivatives :
 - ✓ **sh** : bourne **shell** (obsolete), Traditional, basic shell found on Unix systems, by Steve Bourne.
 - ✓ **ksh** : **korn shell is derived from the sh**.
 - ✓ **zsh** : **Z shell**, quite recent and contains the best ideas from bash, ksh and tcsh.
- **C Shell** and its derivatives :
 - ✓ **csh** : The **C shell** (obsolete), Once popular shell with a C-like syntax. developed by Berkeley University.
 - ✓ **tcsh** : The **TC shell** (still very popular), compatible implementation with evolved features (command completion, history editing and more...)
- **Bash**: The **Bourne Again shell** (most popular), an improved implementation of sh with lots of added features too.
- **fish**: **Friendly Interactive Shell** (a great new shell) : <https://fishshell.com/>

☒ 5 Ways to Find Linux Kernel Version :

- Use the uname Command : **# uname -r**
- Use the hostnamectl Command : **# hostnamectl**
- Print the Contents of /proc/version : **# cat /proc/version**
- Find the Kernel Version With the dmesg Command : **# dmesg | grep Linux**
- Using dpkg Command : **# dpkg --get-architecture | grep linux-image**

➔ Notes !

☒ To work on a **Linux** distribution, you need to have an account, so enter your username (login) and password.

- **Username:** 1BTg**X**b**Y** (**X** is your **Group** number and **Y** is your **Machine** number)
- **Password:** NSCS-s**Z** (**Z** is your section number)

☒ **Attention :**



UNIX/Linux differentiates between uppercase and lowercase. Most commands should be written in lowercase. Always separate the command from its arguments with spaces.

☒ **Remarks:**

- ✓ If the name is too long to type, use the **tab** key.
- ✓ Use the **up** and **down** arrows to scroll through commands.
- ✓ **Log out** at the end of the session. You are responsible for your account.

➔ Objectives

- Learn about the **Linux environment (Ubuntu)** in graphical and console (terminal) mode.
- Discover the **terminal** and the **shell** (run commands for the first time):
 - ✓ Learn what a **command** is ? **Command Prompt** ?
 - ✓ Learn the **type** of a command.
 - ✓ Learn **how to get help** on a command.
- ...

PART 1: THEORETICAL PART / COURSE QUESTIONS*Course Questions ...***PART 2: ACTIVITIES**★ **ACTIVITY 1:** *Using the command **man***

Construct a table according to the model given, using the following commands (with or without options)

- **man**, **man -k**,
- **cd**, **cd ..**, **ls**, **ls -a**, **ls -l**, **ls -al**, **pwd**,
- **mkdir**, **rmdir**, **cp**, **cp -R**, **mv**, **rm**, **rm -R**,
- **touch**, **echo**, **history**, **history -c**,
- **who**, **passwd**,
- **cat**.

For each of these commands, study the help page and give a description of the command. Test each of these commands on real examples and analyze the system response.

If you are unfamiliar with the use of a command, searching for it using the command **man** should become a reflex. It is strongly recommended that you keep this table and add to it as you go.

Command	Description Syntax Example
man	Return the user manual if the command is available. man [options] nom_commande ex : man man
man -k	Lists the commands associated with the given keyword. man -k mot_cle ex : man -k copy
cd	
...	

★ **ACTIVITY 2:** *Basic use of **Shell***

1- Basic Commands : Review the use of the following commands : **cd**, **mkdir**, **ls**, **rmdir**, **rm**, **cp**, **man**, **date**, **pwd**, **mv**, **echo**. For each command, describe in a sentence what it does and indicate the role of the options given in square brackets (you should know these options by heart):

- ↪ **cd**
- ↪ **mkdir** [-p]
- ↪ **rmdir**
- ↪ **pwd**
- ↪ **man**
- ↪ **ls** [-l] [-a] [-R] [-1]
- ↪ **rm** [-i] [-r]
- ↪ **cp** [-i] [-r] [-a]
- ↪ **mv** [-i]
- ↪ **date**
- ↪ **echo** [-n]

2- First steps: towards long-term effectiveness

Let's create a working directory for the **OS1** course, and a sub-directory for this **practical work**. To save time during the various practical sessions, organise your files well. You may find it useful to have a personal backup on a USB key.

You should have the following directories in your login directory:

- **bin** : personal scripts and commands ;
- **tmp** : temporary tests, to be deleted regularly.
- One directory per module, containing a subdirectory per **TP** (TP01, TP02, ...: exercises of TP 1, 2 ...).

Don't leave files in your connection directory. Use subdirectories. **This will make you more efficient in the long term.**

3- Command man (or xman)

You can search for a **keyword** interactively when viewing the manual for a command (the command **man** uses the command **less** to display the documentation). The search is launched by pressing the / key (see the **less** command manual for more details).

- Look up the word '**pattern**' in the de **less** manual.

4- Command ls

Using the command **ls** and its various options (see **man ls**), view the contents of your current directory as follows:

1. Simple list.
2. List showing hidden files (those whose names begin with "."). Note the presence of the 2 entrances "." and "..".
3. List with full description of each reference (permissions, number of links, dates, size, user, group, etc.).
4. List with a full description and a more comprehensible file size format.
5. Recursive list (moves down sub-directories).
6. List in chronological order (the command **touch** can be used to change the modification date of a file).
7. List by access date instead of creation date. To see a change, use the command **cat** 'filename' to change the last access date.
8. Simple list of contents with display of file type (directory /, symbolic link @, executable *).
9. List with **inode** number . (you can check this by using the command **ln** to create a physical link to an existing file).

✪ ACTIVITY 3 : Using the Command ls

Using the various options in the command **ls** (use **man ls**), give the command for each of the following cases to obtain the required result:

1. List all elements in the current directory.
2. List all elements, including hidden ones, in the current directory.
3. Display a full description of the permissions, number of links, dates, etc. in the current directory.
4. Display a full description with a more comprehensible file size format.
5. Display a list of all files and folder contents (recursive list).
6. Display a list in chronological order (by date of modification).
7. Display a list by access date.
8. A list of all elements sorted in descending order of size.
9. A list with a full description of all the elements in the current directory, sorted in descending order of size.
10. A list of the types of all the elements in the current folder.

★ ACTIVITY 4 : *Internal & External Commands*

There are **two types of system command**: **internal** commands and **external** commands. Internal commands are commands whose programs reside in working memory, while external commands are loaded into memory when they are called.

One way of distinguishing between them is to use the command **which** followed by the name of the command to be tested. If the command returns a path to an executable file, then the command is external. Otherwise (and if the command exists) the command is internal.

1. Recall the principle of executing the command **which**.
2. Why use internal commands ?
3. Give examples of both types.

★ ACTIVITY 5 : *Text-based session*

This type of session gives access to the same tools as the console in graphics mode, but does not allow graphics applications to be launched.

1. Open a session in **text-based mode** using the adequate key combination.
2. Enter your account details: login and password.
3. Run and assign the following commands :

hostname	
uname -a	
id	
pwd	

4. Return to the graphical session using the appropriate key combination.
5. Open another session in text mode using a different key combination from the first.
6. Return to the first session using the corresponding key combination.
7. Run the command **who** and explain the result.
8. Close both sessions in text-based mode using the command **exit**.

★ ACTIVITY 6 : *The Alias*

Aliases can be used to define **shortcuts** for common or repetitive commands.

1. The command to define your own **alias**:

```
alias my_command = 'full_command'
```

2. Run the command **alias**.
3. Create a command **lsa** that allows you to list the contents of a directory, including hidden files.
4. Run the command **lsa**.
5. Run again the command **alias** and notice the difference.
6. Create a command **lc** that allows to activate the colored display.
7. Run the command **unalias lsa**.
8. Run the command **alias**.
9. Determine the role of the control **unalias**.